

SKP16C1N2

USERS MANUAL **Ver. 1.0**

Table of Contents

1.0 Introduction	2
2.0 Contents of Product Package	3
3.0 Limited Guarantee and Support.....	4
4.0 System Connectivity	5
4.1 Host Computer Requirements	5
4.2 SKP16C26A Board.....	5
4.3 ICD (RTA-FoUSB-MON)	5
4.4 Software Development Tools.....	6
4.4.1 HEW (High-performance Embedded Workshop)	6
4.4.2 NC30WA Entry Version	6
4.4.3 KD30 Debugger.....	6
4.4.4 FoUSB (Flash-over-USB™) Programmer	6
5.0 Hardware.....	7
5.1 SKP16C26A Board.....	7
5.2 SKP16C26A Board Block Diagram.....	7
5.3 M16C/26A Group of MCUs.....	8
5.4 SKP16C26A Board Jumper Configuration	9
5.4.1 JP1: MCU (U1) Power	9
5.4.2 JP2: Xout.....	9
5.5 LCD (Liquid Crystal Display).....	9
6.0 System Operation & Limitations	10
6.1 Kernel (ROM Monitor) Introduction	10
6.2 Pin and Peripheral Limitations	10
6.3 Memory Map	11
6.4 Register Operation Limitations	11
6.5 Limitations on Interrupts	11
6.6 Stop or Wait Mode Limitations.....	12
6.7 User Program's Real-Time Capability.....	12
6.8 Performing Debug Using Symbols.....	12
7.0 SKP16C26A Board Specifications.....	13
7.1 Hardware Specifications.....	13
7.2 Power Supply Requirements	13
7.3 Operating Environment.....	13
Appendix A. Troubleshooting Guide	14
Appendix B. Reference Manuals	17
Appendix C. Expansion Headers.....	18
Appendix D. SKP16C26A Schematics.....	19
Appendix E. SKP Board Dimensions.....	20
Appendix F. RTA-FoUSB-MON (ICD)	21

1.0 Introduction

The SKP16C1N StarterKit Plus (SKP) is a low-cost development environment for evaluating M16C/26A group microcontrollers (MCU) and Renesas software development tools. The kit comes with a complete software development tool chain including, HEW (IDE, GUI), NC30WA (C-compiler, assembler, and linker), KD30 (Debugger), and FoUSB (Flash-over-USB™ Programmer).

A real-time, source-level debug environment is implemented using the KD30 debugging software with the RTA-FoUSB-MON Flash Programmer/In-Circuit Debugger (ICD). The Flash-over-USB™ (FoUSB) Programmer software, with the ICD, allows in-system programming of the M16C/1N flash MCU.

The ICD and firmware provide a convenient USB (Universal Serial Bus) interface between the SKP16C1N board and the host PC. This interface reduces resource requirements on the M16C/1N MCU, allows faster code downloads and, can also be used with many other Renesas Flash MCU's, SKP's, and user's target board.

2.0 Contents of Product Package

This section describes the contents of the SKP16C1N product package. When unpacking your SKP16C1N, please check to see that all products listed below are included.

2.1 SKP16C26A StarterKit Plus Product List

Table 2-1 lists the products included in the SKP16C26A.

Table 2-1 SKP16C26A Product List

Product Name	Quantity	Remark
SKP16C1N Board	1	M16C/1N SKP Board
RTA-FoUSB-MON (ICD)	1	KD30 Debugger/ FoUSB Programmer Interface Device
6" 10-Pin Target Cable	1	Connects SKP16C26A Board (J5) and the ICD
6' Mini USB Cable	1	Connects ICD to the Host PC
9 VDC Wall adapter	1	Power Supply for SKP
SKP CD-ROM	1	Auto-install program HEW (IDE) NC30WA (C-compiler, assembler, and linker) KD30 Debugger FoUSB Programmer Manuals Tutorials Sample programs

2.1.1 SKP CD-ROM

The CD-ROM contains the electronic manuals and software necessary for developing programs. Your computer must have Netscape Navigator® or Microsoft®'s Internet Explorer to view the help files and Acrobat Reader to view the manuals.

Insert the enclosed CD into your computer and SKP installer will auto-start. The SKP installer program will create a C:\HEW3 and C:\MTOOL folder on your machine. All development tools (HEW, NC30WA, KD30 Debugger, and FoUSB Programmer) can be found under C:\HEW or C:\MTOOL. Documentation, sample code, and other SKP related files are in the C:\MTOOL\SKP16C1N folder.

If the SKP installer program does not start up, browse the CD's root folder and double-click on '**skp_installer.exe**' to start installation.

3.0 Limited Guarantee and Support

Renesas Technology America, Inc., warrants the SKP16C1N2 to be free from component or assembly defect for a period of 180 days from the date of purchase. Settlement is limited to repair or replacement of the product only. Renesas Technology America, Inc., does not assume any liability arising out of the application or use of any product, circuit or procedure described herein. No other liability or warranty apply, expressed or implied. Software warranty is limited to replacement of the CD only. While every attempt has been made to ensure accurate documentation, Renesas Technology America, Inc., cannot be held responsible for error or omissions and reserves the right to make changes without further notice.

“Flash-Over-USB” is a trademark of Renesas Technology America, Inc. All trademarks are the property of their respective owners.

4.0 System Connectivity

The following lists the hardware and software products required for using the SKP16C26A StarterKit Plus.

- Host Computer (supplied by user)
- SKP16C1N Board
- RTA-FoUSB-MON (ICD)
- Mini USB Cable
- Target Cable
- Wall adapter
- Software Tools (HEW IDE, NC30 Compiler/Linker, KD30 Debugger, FoUSB Programmer)

Figure 4-1 shows the system connectivity for the SKP16C1N.

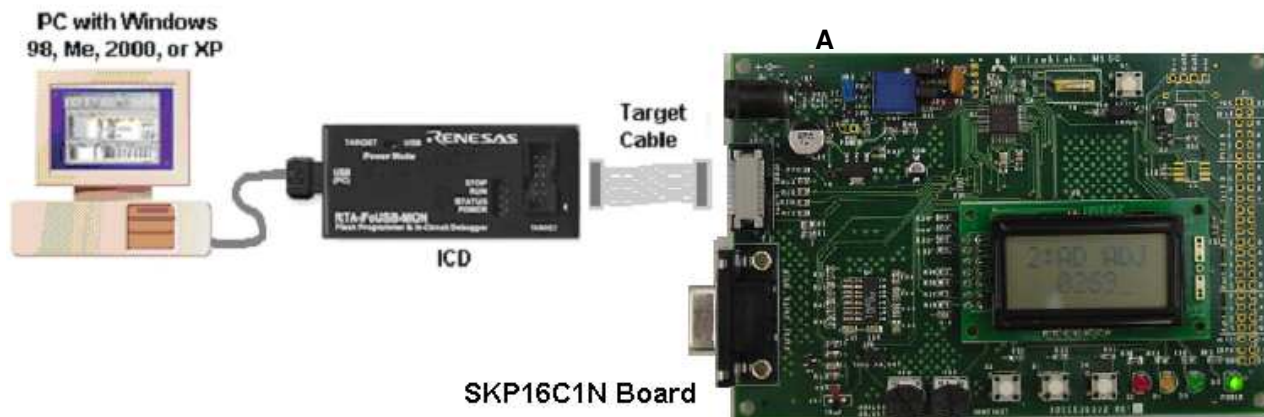


Figure 4-1 SKP16C1N System Connectivity

4.1 Host Computer Requirements

The minimum requirement to be able to use the software that comes with the SKP16C26A is a PC with a USB port and Microsoft Windows 98, ME, 2000, or XP.

4.2 SKP16C1N Board

The SKP16C1N board provides an evaluation and development environment for the M16C/1N group MCUs. See section 5.0 'Hardware' for more details.

4.3 In-Circuit Debugger (RTA-FoUSB-MON)

The ICD provides a plug-and-play debugging and programming interface to the SKP16C1N board via the host computer's Universal Serial Bus (USB). The USB port also provides power to the SKP16C1N board and ICD thereby eliminating the need for an external power supply.

4.4 Software Development Tools

The SKP installer program installs all the development tools. For details on installation, see the Quick Start Guide or instructions in the Appendix of this manual. A brief description of all the included tools follows. Please refer to the individual Tool manuals for detailed information.

4.4.1 HEW (High-performance Embedded Workshop)

HEW provides a Graphical User Interface (GUI) that integrates the software development tools and includes the C-compiler, assembler, linker, and editor.

4.4.2 NC30WA Entry Version

The NC30WA Entry version C-compiler included in the SKP has no time expiration, but has the following limits:

1. The software comes without warranty.
2. The compiler always compile with “-fansi” option.
3. The compiler optimization settings is fixed and cannot be changed
4. The ‘-finfo’ option for other software tools such as stack viewer, etc. is not supported.

If your development needs exceed the above limitations, please go to the following web site http://www.renesas.com/eng/products/mpumcu/toolhp/datsheet/m16c_e/nc30wa.htm and download the full version (4 calendar month trial).

4.4.3 KD30 Debugger

KD30 is a remote debugger that runs on the host PC. While communicating with a kernel (i.e. a ROM monitor program) on the target MCU through the ICD, KD30 provides a highly efficient evaluation environment. KD30 features include:

- Source-line debug for assembly language, structured assembly language, C language
- Run command with **2** breakpoints* for the M16C/1N
- RAM monitor function
- C variable “watch” window

Note: The number of breakpoints will vary depending on the M16C flash MCU used.

4.4.4 FoUSB (Flash-over-USB™) Programmer

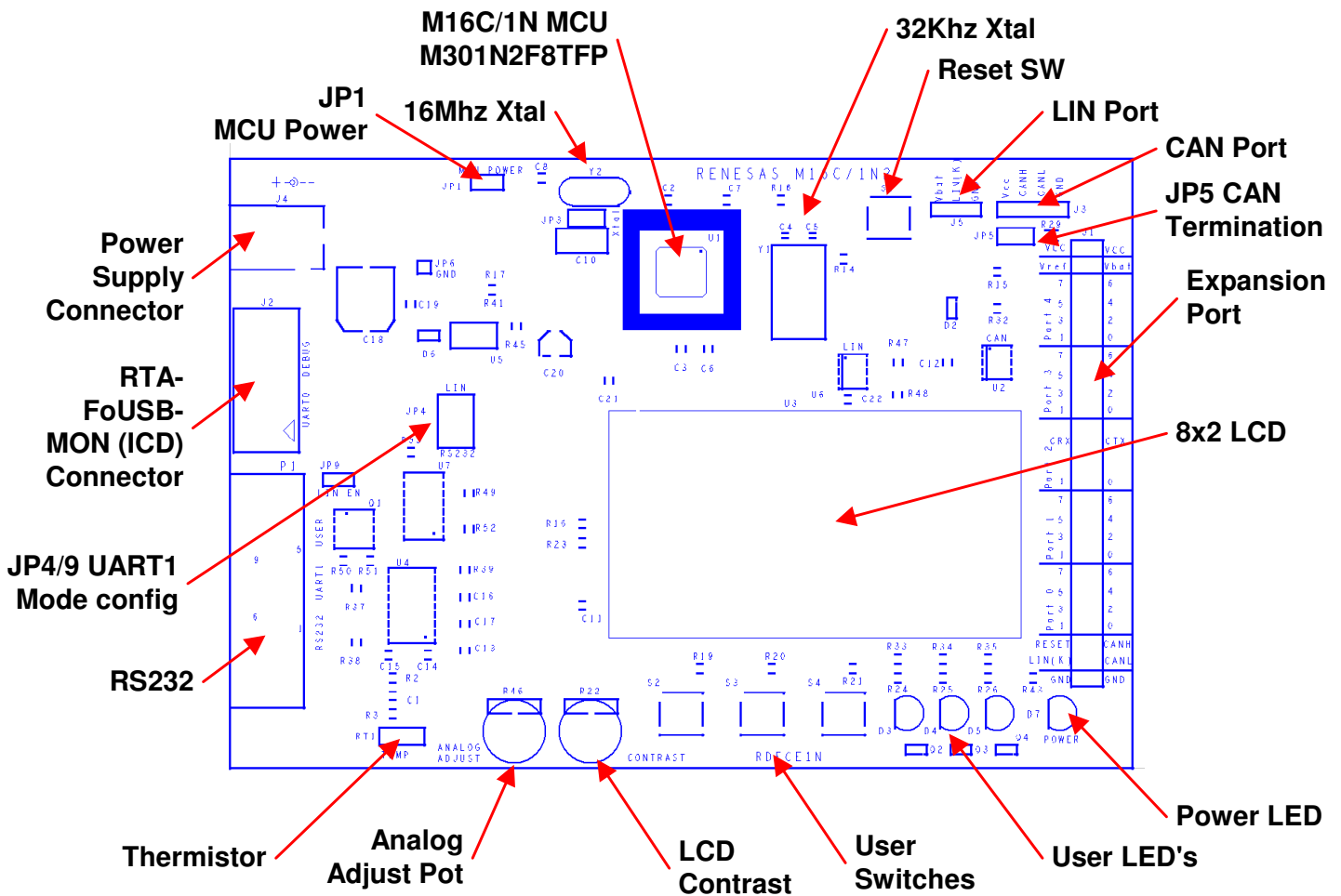
The Flash Over USB Programmer application provides In-System Programming capability for the starter kit or any target board using an M16C family flash MCU (i.e.: R8C, M16C, M32C). Please see the RTA-FoUSB-MON Users Manual for more details.

5.0 Hardware

5.1 SKP16C1N Board

Figure 5-1 shows the SKP16C1N Board with major components identified.

Figure 5-1. SKP16C1N Board



5.2 SKP16C1N Board Block Diagram

The SKP16C1N board incorporates an M301N2F8TFP (48-pin QFP) from the M16C/1N group of microcontrollers designated as U1. Figure 5-2 shows the SKP16C1N block diagram.

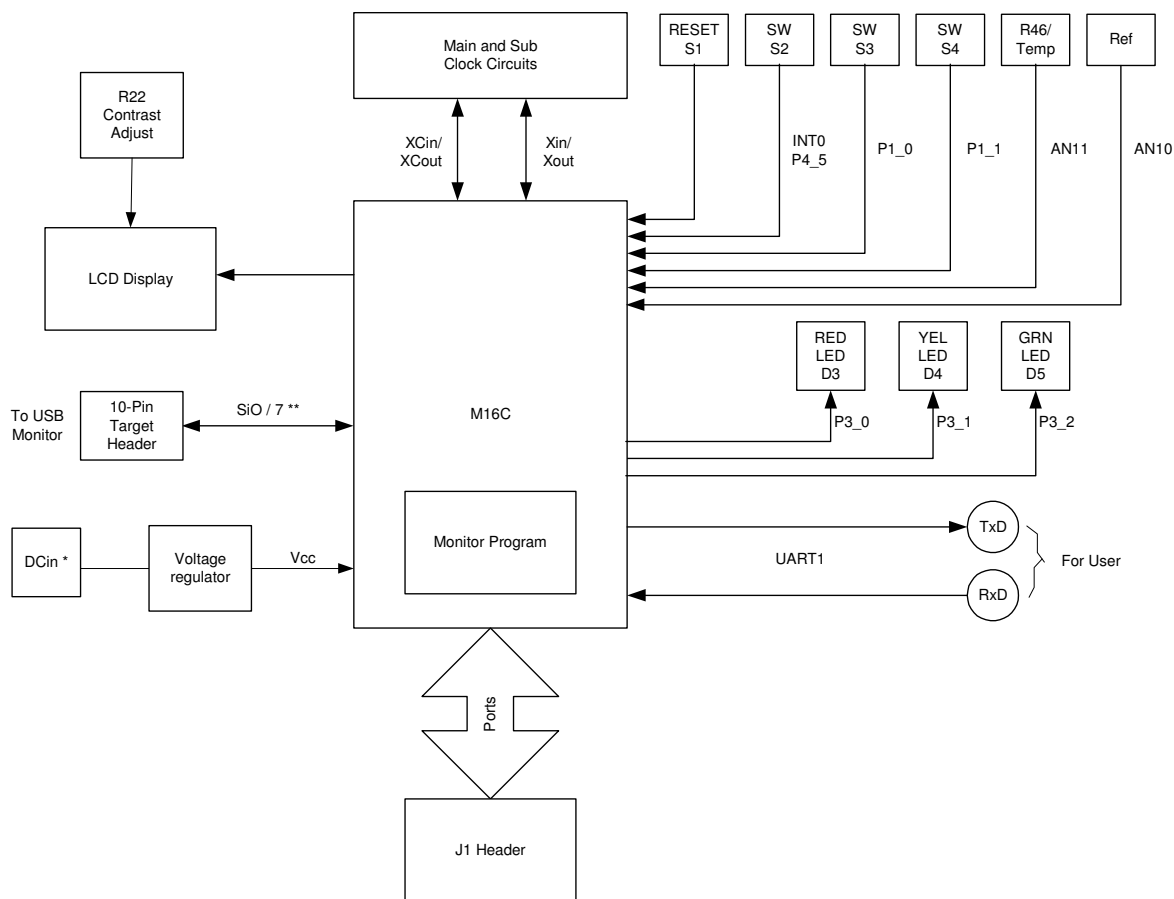


Figure 5-2. SKP16C1N Board Block Diagram

5.3 M16C/1N Group of MCUs

The M16C/1N group of 16-bit single-chip, flash microcontrollers (MCU) is part of the M16C/10 family and utilizes the M16C/60 series CPU core. The hardware and software manuals for the M16C/1N group of microcontrollers can be found under C:\MTOOL\SKP16C1N\Docs folder in your PC or from the Start menu (Start > Programs > RENESAS-TOOLS > SKP16C1N > Document Descriptions) after SKP software installation.

5.4 SKP16C26A Board Jumper Configuration

5.4.1 JP1: MCU (U1) Power

JP1 is used to connect the Vcc pins of the M16C/26A MCU to the Vcc/MCU Power of the board. It can be used to measure current/power consumption of the MCU during various modes of operation. For normal operations, JP1 must be shorted.

JP1 is shorted by default.

5.4.2 JP3: Xout

JP2 is used to connect the output of the MCU's internal amplifier to the crystal for oscillation. It can be used to disconnect Xout when an externally driven clock is connected to Xin. For normal operations, JP3 must be shorted.

JP3 is shorted by default.

5.4.3 Default Jumper Settings

Jumper	JP1 MCU Power	JP3 Xout
Setting	Shorted	Shorted

5.5 LCD (Liquid Crystal Display)

The LCD is a 2-line by 8-character display with a KS0066 controller IC.

6.0 System Operation & Limitations

The SKP16C1N provides sophisticated debugging features at a low cost but it does have some limitations when used with the KD30 Debugger and ICD. Section 6.1 introduces the kernel (ROM monitor) program and its purpose. The limitations when this kernel is running with the user program are listed in table 6-1.

Table 6-1 System Limitations (when used with KD30)

Item	Please Refer To
User Limitations	6.2 Pin and Peripheral Limitations
	6.3 Memory Map
	6.4 Status After Reset
	6.5 Register Operation Limitations
	6.6 Limitations on Interrupts
Debugger Limitations	6.7 Stop or Wait Mode Limitations
	6.8 User Program's Real-time Capability

6.1 Kernel (ROM Monitor) Introduction

During debug (used with the KD30 debugger), a small program, called a kernel, is downloaded to the M16C/1N. The kernel communicates with the KD30 Debugger through the ICD regarding MCU status during user code debugging operations.

There are no special steps required in the user program to make use of the ICD. The operation of the kernel is transparent to the user but there are some limitations and these are discussed from section 6.2.

After starting KD30, the ICD downloads the kernel to the M16C/1N if it does not exist (e.g. blank device or programmed with FoUSB Programmer). After downloading the kernel, KD30 opens the Program Window and the M16C/1N is ready for downloading user code.

Connecting the ICD without starting KD30 will not affect the lines connected between the ICD and the M16C/1N – the ICD keeps the lines in high-impedance state. The ICD only drives the pins after KD30 or FoUSB Programmer is started.

After program debug and verification, you can then create and download a binary, Intel (.hex) or Motorola (.mot), file to the M16C/1N using the FoUSB Programmer. The program operation erases the kernel and leaves the user program only.

6.2 Pin and Peripheral Limitations

SIO/UART0 pins are used for communication between the M16C/1N kernel on the SKP16C1N board and KD30 Debugger through the ICD. Do not connect these pins to any other circuit, as UART0 cannot be used in the user program. For details, please see the RTA-FoUSB-MON (ICD) User Manual on Target M16C ROM Monitor Resources or related ICD application notes.

6.3 Memory Map

The amount and memory locations of the kernel, to communicate with the KD30 and ICD, are using on the M16C/1N on the SKP16C1N board is shown on Figure 6-2.

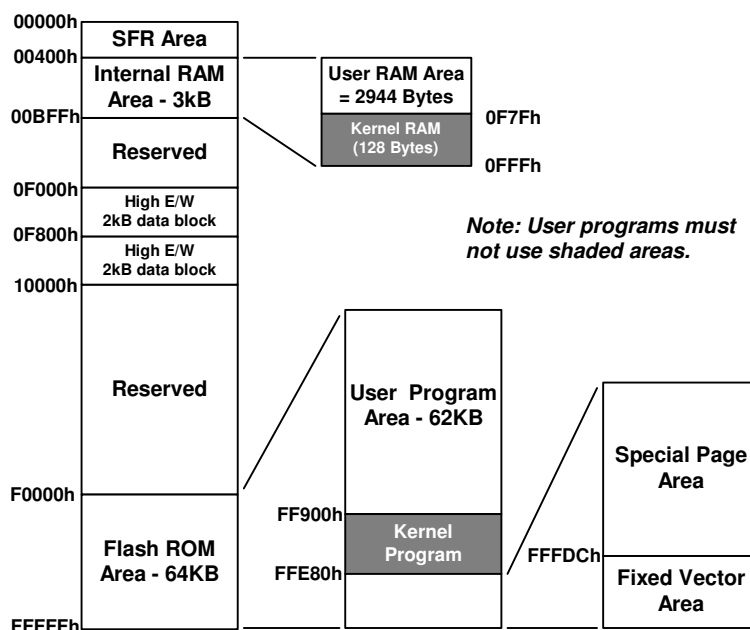


Figure 6-2 M301N2F8TFP Memory Map with the Kernel Program

Note: The kernel occupies memory associated with special-page vector numbers 18-19 and 192-255. The user reset vector is re-mapped to address FFFD8h by the kernel.

6.4 Register Operation Limitations

Table 6-3 lists the limitations on register operation. The registers are inhibited from any modification. If register contents are modified in any way, kernel operation cannot be guaranteed.

Table 6-3. Limitations on Register Operation

Register Name	Restriction
User Interrupt Stack Pointer	Range 0F80H – 0FFFFH is used by the kernel
UART0 Transmit/Receive Mode Register	Do not change
UART0 Transmit/Receive Control Register 0	
UART0 Transmit/Receive Control Register 1	
UART0 Interrupt Control Register 0	Do not change
UART Transmit/Receive Control Register 2	Do not change bits 0 and 2
UART0 Transmit Buffer Register	Do not write to this register
UART0 Receive Buffer Register	Do not read this register
Port 1 and Port 1 DDR	To prevent changes on P1_7 data and direction, use read-modify-write only instructions (BSET, BCLR, AND, OR, etc)

6.5 Limitations on Interrupts - Vectors that Reside in the Hardware Vector Table

Table 6-4 lists the limitations on hardware interrupt (i.e. fixed) vector addresses.

Table 6-4. Interrupt Vector Addresses

Interrupt Cause	M16C/26A Vector Address	Kit Specification
Undefined	FFFDCH ~ FFFDFH	User available
Overflow	FFFE0H ~ FFFE3H	User available
BRK Instruction	FFFE4H ~ FFFE7H	User inhibited
Address Match	FFFE8H ~ FFEFBH	User inhibited
Single-step	FFFECH ~ FFFE FH	User inhibited
Watchdog Timer	FFFF0H ~ FFFF3H	User available (Note 1)
DBC	FFFF4H ~ FFFF7H	User inhibited
NMI	FFFF8H ~ FFFF BH	User available
RESET	FFFFCH ~ FFFF FH	Reset vector (Note 2)

NOTES:

- (1) The Watchdog Timer vector is shared with the oscillation stop interrupt. The vector is available for oscillation stop interrupts, but avoid using the vector for watchdog timer interrupts.
- (2) The kernel transparently relocates the Reset vector to FFFD8H.

6.6 Stop or Wait Mode Limitations

The kernel cannot be run in STOP or WAIT modes. Do not use these modes when debugging your program.

6.7 User Program's Real-Time Capability

Please be aware that while the kernel is in a "STOP" state, the hardware peripherals will continue to run. Therefore, interrupts may not be serviced properly. Also, the watchdog timer will not be serviced and will likely time out if active.

While the kernel is in a "RUN" state, there is no overhead on the application code, UNLESS a RAM monitor window is open in KD30. This window requires periodic communication with the MCU. This communication suspends normal application operation while servicing the request (approximately 2000 BCLK cycles for each 16 bytes of data displayed in the window are used per window update). The user must determine whether or not this behavior is acceptable.

6.8 Performing Debug Using Symbols

Normally when a new project is created using HEW, debugging symbols are enabled. If you are unable to debug your program using symbols in KD30, add the debug option [-g] in HEW before compiling the programs. To enable the [-g] option, perform the following:

- Open the workspace and project in HEW.
- Select [Renesas M16C Standard Toolchain] from Options pull-down menu.
- Click on Link tab.
- Select [Output] under the [Category] list box.
- Click on checkbox for [-g] 'Outputs source debug information....'
- Click on [OK] button

For more information, see the HEW user's manual.

7.0 SKP16C1N Board Specifications

7.1 Hardware Specifications

Table 7-1 lists the specifications of the SKP16C1N Board.

Table 7-1. SKP16C26A Board Specifications

Item	Specification
MCU	M301N2F8TFP
Clocks	Main Clock: crystal 16MHz, or ring oscillator Sub Clock: 32.768kHz crystal
Memory (with KD30 + ICD)	RAM: 3kB (2944 Bytes) High E/W Data Block: 2kB x 2 Flash ROM: 62kB (63,744 Bytes)
Connectors	[J2]: Serial interface connector (UART0 for KD30 + FoUSB-ICD) [J1]: 50 pin (user supplied) headers (for user target connection) [J3]: CAN network connection [J5]: LIN network connection [P1]: RS232 to UART1 [J4]: Power Supply connection
Jumpers	[JP1]: MCU Power for Icc Measurements [JP3]: Xout to main crystal Y2 [JP4]: UART1 configuration [JP5]: CAN 120 ohm termination [JP9]: LIN communications enable
Switches	[S2]: pushbutton (connected to P4_5) [S3]: pushbutton (connected to P1_0) [S4]: pushbutton (connected to P1_1) [S1]: pushbutton (connected to Reset)
LED's	[D1] (Red): User output (connected to P3_0) [D2] (Yellow): User output (connected to P3_1) [D3] (Green): User output (connected to P3_2)
LCD	2-line x 8-character LCD with KS0066 controller IC

7.2 Power Supply Requirements

The SKP16C1N Board will draw ~50mA. With the ICD, the current draw will be about 100mA.

7.3 Operating Environment

Table 7-2 lists the environmental conditions for using and storing the SKP16C1N board. When storing the board, place it in a conductive bag and then in the packing box your product was shipped in from the factory.

Table 7-2. Operating Environment

Environmental Condition	Ambient Temperature	Ambient Humidity
Operating	0 - 55°C (No corrosive gas allowed)	30 to 80% (non-condensing)
Storage	-30 to 75°C (No corrosive gas allowed)	30 to 80% (non-condensing)

Appendix A. Troubleshooting Guide

This section discusses possible problems you may encounter while installing the software (and drivers) and while running the KD30 Debugger or FoUSB Programmer applications. This section also discusses the countermeasures and solutions to resolve these problems.

If, for any reason, you cannot resolve the problem, please contact your Renesas representative for assistance.

A.1 USB Driver Problems

This section discusses the usual problems with the driver installation and how to fix it. The most common problem encountered is that Windows did not properly install the driver and so the ICD is not recognized. This may also cause the device status to indicate that the device is not working properly. An indication of this problem is the ICD status yellow LED - it blinks about 2-3 times a second. When the driver is installed properly, the yellow LED should only blink every second.

Before trying the following steps, try restarting your PC and see if this resolves the problem. You can check the status using the Device Manager. If the ICD appears under the Universal Serial Bus Controllers with NO red X or yellow exclamation point, the driver was installed properly.

NOTE: If you are using Windows 2000 or XP, you need Administrator privileges to install the drivers.

For cases where the 'Device Status' states the device is not working properly, please try the following:

- Double-click on 'Renesas USB-Monitor' and a Renesas USB-Monitor Properties dialog box appears.
- Click on 'Driver' tab and click on 'Update Driver' button.
- Select 'Display a list...' and click on 'Have Disk' button.
- Specify and locate the 'C:\MTOOL\FoUSB\USB Drivers' folder on your PC and install 'usbmon.sys' driver.

If you encounter problems on installing the drivers, you can try the following.

(1) Windows 2000

- i. Copy the **fousb.inf** and **usbmon.inf** files from **\USB Drivers** folder to \WINNT\INF folder.
- ii. Copy the **fousb.sys** and **usbmon.sys** files from **\USB Drivers** folder to \WINNT\SYSTEM32\drivers folder.

(2) Windows 98 or XP

- i. Copy the **fousb.inf** and **usbmon.inf** files from **C:\MTOOL\FoUSB\USB Drivers** folder to \WINDOWS\INF folder.
- ii. Copy the **fousb.sys** and **usbmon.sys** files from **C:\MTOOL\FoUSB\USB Drivers** folder to \WINDOWS\SYSTEM32\drivers folder.

A.2 KD30 Problems

This section discusses the cause of the problem and countermeasures to resolve it. The common problems encountered with KD30 are:

- USB option cannot be selected from the initialization screen
- Cannot connect to target
- KD30 already exists
- Issues that may come up during debug operations

A.2.1 USB Option Unavailable

When USB cannot be selected from KD30's Init dialog box, you might be using an old version of KD30 that does not support USB. Uninstall this version of KD30 and install latest version of KD30 from the \Tools\KD30 directory of the SKP CD.

A.2.2 Cannot Connect to Target

When the message 'Can't connect with the target' is displayed after KD30 startup, there are several reasons that may cause this message to appear. Each cause and the corresponding countermeasure are discussed below.

- The SKP is not connected correctly.

Please connect the ICD to your PC. Note: Regardless of whether the ICD is bus or target-powered, please connect target board to ICD board first before plugging the USB cable to your PC. Please see section 4 on system (SKP16C26A) connectivity.

- The ICD has no power (Power LED on ICD is off).

Please ensure that the Power Mode switch on the ICD is on the 'USB' side. In 'Target' position, the ICD is powered from the target side.

- USB was not selected on the Init dialog box.

Please select 'USB' from the Init dialog box that is displayed right after you start KD30.

- The selected MCU on the ICD board and the actual target MCU (M16C/26A) do not match.

Close the error message by clicking on 'OK' button, and then click on the 'Cancel' button of the KD30 Init window to close KD30. Make sure you select 'M30280FA.mcu'. If the MCU loaded on the ICD is different, KD30 will re-program the ICD to match it.

- The target MCU is damaged.

Try a different target board and see if KD30 will come up as you may have a damaged board or MCU.

A.2.3 KD30 Already Exists

When a message 'KD30 already exists' is displayed, the usual cause of this problem is that the KD30 application was not properly closed. Please unplug the USB cable first and wait a few seconds. Then check if KD30 is already running by looking at your task bar. If KD30 cannot be found there, bring up Task Manager (press CTRL-ALT-DEL once and click on 'Task Manager' button). Select KD30.exe on the 'Processes' list and click on 'End Process' to terminate KD30. Then re-plug in the USB cable and restart KD30.

A.2.4 Issues that may come up During Debug Operations

While using KD30 to debug user code, some issues may come up because the limitations discussed in section 6 were not satisfied. The common issues are listed on table A.2, including the countermeasures.

Table A.2. Problems while using KD30

Problem	Possible Cause/s and Solution
After stepping a few instructions, KD30 hangs	<ul style="list-style-type: none"> Changes were made to UART0 SFR's. Do NOT change UART0 SFRs in user program.
Breakpoints do not seem to work	<ul style="list-style-type: none"> KD30 is in "FreeRun" mode. Change the RUN mode to "Sampling" from the "Init" window (Environment > Init).
KD30 locks up (cannot stop program) or Communication error message is displayed.	<ul style="list-style-type: none"> Changes were made to UART0 SFR's. Do NOT change UART0 SFRs in user program. Ensure no limitations in Section 6 were violated. Re-initialize the system without closing KD30. See note below. Do a hardware reset. User-program runaway may be corrupting the kernel (RAM, interrupt vectors, flags, etc.) Close KD30, hit reset button on the SKP16C26A to reset the board, and then restart KD30.
Download problems	<ul style="list-style-type: none"> Filenames or directory names contain spaces or special characters. HEW project not properly set up (startup files missing or out of order, files added to wrong member, etc). Try creating a new project and adding your source files to it. For details, please see HEW user's manual.

To re-initialize the system without closing KD30, try the following:

- Press the [OK] button on the error dialog box to close it.
- When an Exit dialog box appears, press the [Cancel] button to close it.
- Hit reset button on the SKP16C26A board.
- Press KD30 reset button.

After initialization, debugging can resume. However, it is recommended that you download your program again before debugging.

Note: If it has been identified that there are problems with the ICD, please see the ICD's (RTA-FoUSB-MON) user's manual and troubleshooting section of this manual.

Appendix B. Reference Manuals

Item	Title	Description
1.	SKP16C1N Quick-Start Guide	Gets you up and running with the SKP16C1N.
2.	SKP16C1N User's Manual	Installation and operation guide for the SKP16C1N.
3.	SKP16C1N Board Schematic	Schematic diagram for the SKP16C1N Starter Kit board.
4.	SKP16C1N Board BOM	Bill of materials for the SKP16C1N Starter Kit board.
5.	M16C/1N2 Group Hardware Manual	Operation and Specifications for the M16C/1N2 MCU.
6.	M16C/20/60 Series C-Language Programming Manual	ANSI C-language programming guide for the M16C/20/60 series MCUs.
7.	M16C/20/60 Series Assembler Language Programming Manual	Assembly language programming guide for the M16C/20/60 series MCUs.
8.	M16C/20/60 Series Software Manual	Instruction set manual for the M16C/20/60 series CPU cores.
9.	M16C/20/60 Series Sample Programs	Sample programs and application notes for the M16C/20/60 series MCUs.
10.	HEW User's Manual	High-performance Embedded Workshop User's Manual.
11.	KD30 User's Manual	Easy-to-use source and assembly debugger.
12.	AS30 User's Manual	AS30 Assembler User's Manual.
13.	NC30 User's Manual	NC30 C-Compiler User's Manual.
14.	RTA-FoUSB-MON User's Manual	In-Circuit Debugger / Flash Programmer User's Manual.

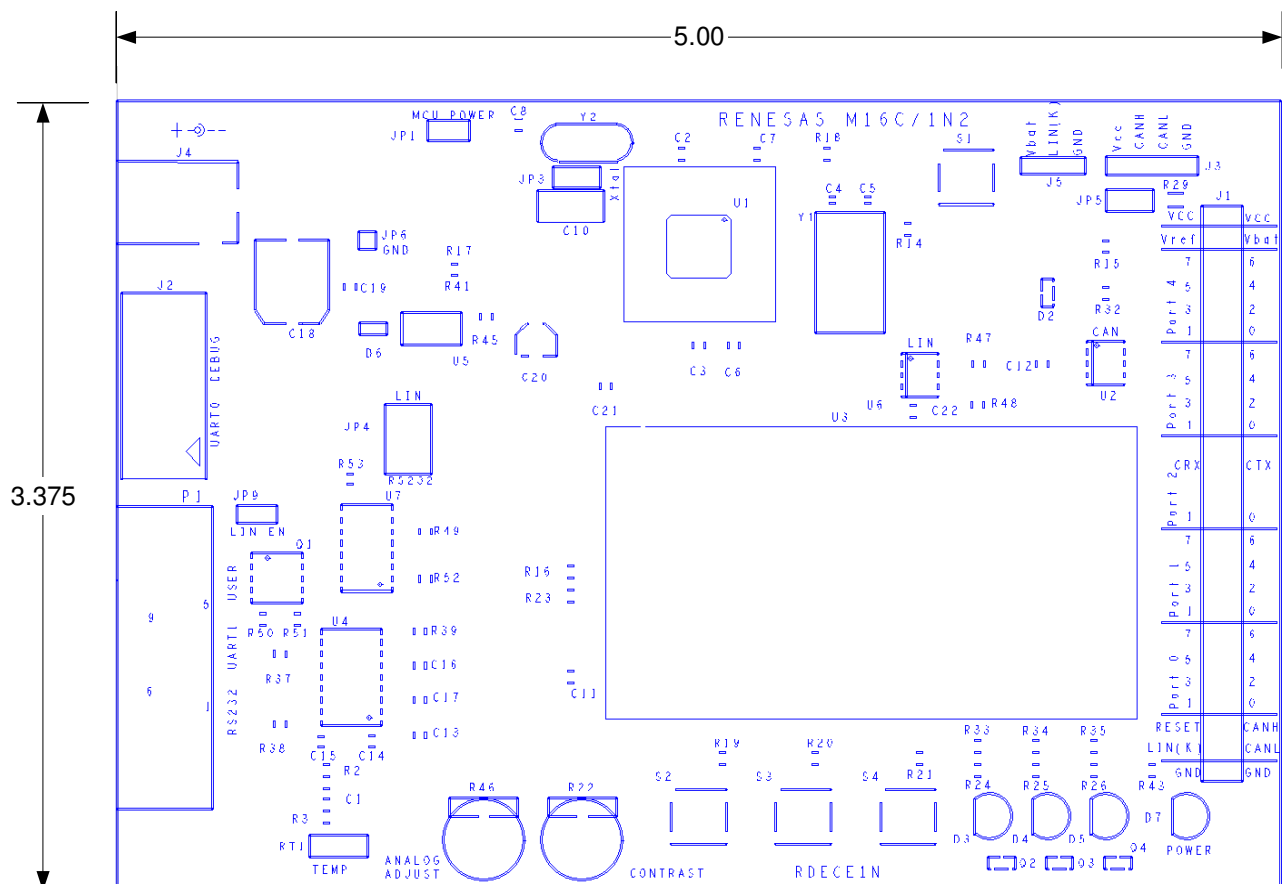
Appendix C. Expansion Headers

J1 Pin	Function	J1 Pin	Function
1	Vcc	2	Vcc
3	Vref	4	
5	P4_7, Xcin	6	P4_6, Xout
7	P4_5, INT0	8	P4_4, INT2
9	P4_3, INT1	10	P4_2, INT3
11	P4_1, ANEX1	12	P4_0, ANEX1
13	P3_7, TXD1, RXD1	14	P3_6, CLK1
15	P3_5, RXD1	16	P3_4, CLKS1,DA
17	P3_3, TCIN	18	P3_2, Tyout
19	P3_1, Tzout	20	P3_0, Txout
21		22	
23	CRX	24	CTX
25		26	
27	P2_1	28	P2_0
29	P1_7, CNTR0	30	P1_6, CLK0
31	P1_5, RXD0	32	P1_4, TXD0
33	P1_3, KI3, AN11	34	P1_2, KI2, AN10
35	P1_1, KI1, AN9	36	P1_0, KI0, AN8
37	P0_7, AN0	38	P0_6, AN1
39	P0_5, AN2	40	P0_4, AN2
41	P0_3, AN4	42	P0_2, AN5
43	P0_1, AN6	44	P0_0, AN7
45	RESET	46	CANH
47	LIN_K	48	CANL
49	Vss	50	Vss

Appendix D. SKP16C1N Schematics

Please refer to “SKP16C1N_schematic.pdf”

Appendix E. SKP Board Dimensions



Appendix F. RTA-FoUSB-MON (ICD)

For details on how to use the ICD, please see RTA-FoUSB-MON User's Manual.